

Rancang Bangun *Pixel Art Converter* Menggunakan Segmentasi berbasis *K-means* *Clustering*

Yuna Sugianela, Nanik Suciati, dan Maulidan Bagus A.R

Departemen Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh
Nopember (ITS)

e-mail: nanik@if.its.ac.id

Abstrak—*Pixel art* merupakan jenis aset grafis yang digunakan pada game. Untuk mengefisiensi pekerjaan pada industri *game*, diperlukan sebuah *converter* citra *raster* biasa menjadi *pixel art*. Tahapan dalam membangun aplikasi *pixel art converter* adalah mengubah warna menjadi aturan yang berlaku pada *pixel art*, lalu membuat tepi gambar menjadi *jaggy* yang merupakan ciri khas grafis *pixel art*. Algoritma segmentasi yang digunakan merupakan segmentasi berbasis *k-means clustering*. Segmentasi ini berguna untuk mengubah warna citra menjadi lebih sederhana. Citra hasil dari segmentasi *k-means* kemudian diolah menjadi citra yang memiliki tepian bergerigi atau *jaggy* yang merupakan ciri khas utama dari *pixel art*. Aplikasi dinyatakan telah memenuhi kebutuhan Tim Desain Maulidan Games. Kualitas citra hasil *pixel art converter* dipengaruhi oleh ukuran citra input, parameter nilai *K* untuk *k-means clustering* serta skala grid untuk membuat tepian *jaggy*. Nilai optimal yang digunakan untuk membuat *pixel art* yang baik yaitu, untuk citra *gradient color* nilai optimal nilai *K* untuk *K-means* adalah 25, sedangkan untuk citra *flat color* menggunakan nilai *K* 16, nilai skala grid untuk membuat tepian adalah 80, ukuran citra optimal adalah 100 x 100 piksel.

Kata Kunci—*pixel art*, segmentasi, *k-means*, *jaggy*.

I. PENDAHULUAN

SALAH satu industri *game* digital di Surabaya yang sedang berkembang adalah Maulidan Games. Saat ini Maulidan Games sedang mengerjakan *Framework Project* yang diberi nama Diamond. Diamond Development Engine memiliki kemampuan dari semua *engine internal*, mulai dari *speech recognition*, *framework* untuk *game* dengan tampilan isometris, manajemen target perhari, template *game*, hingga fitur yang memungkinkan *engine* memprogram *game* sendiri.

Salah satu fitur dari Diamond yang diharapkan adalah sebuah konverter grafis dua dimensi menjadi bentuk *pixel art*. Grafis yang dimaksud adalah aset-aset *game* berupa desain karakter, tombol-tombol, background dan lain sebagainya. *Pixel art* merupakan jenis desain yang digunakan pada game tahun 1990-an seperti Pacman, Mario Bros, Galaga, dan lain lain. *Pixel art* adalah karya seni digital yang diciptakan melalui grafis *raster software* yang diedit pada *pixel level*. *Pixel art* pertama kali ditemukan oleh Adele Goldberg dan Robert Flegal dari Xerox Palo Alto Research Center pada tahun 1982. [1]

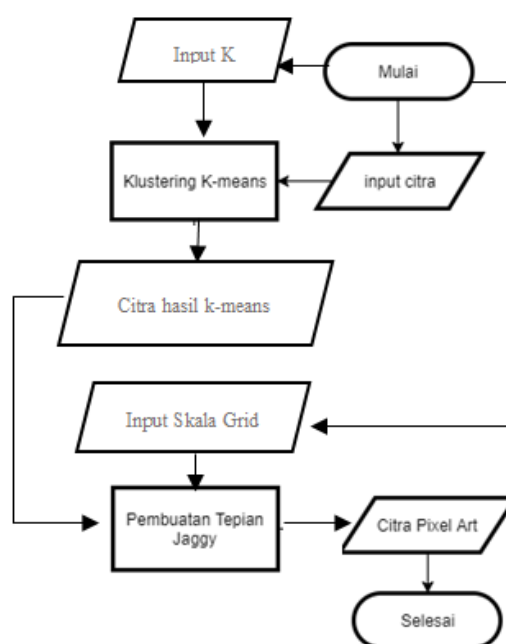
Harapan dari dikembangkannya *Pixel Art Converter* pada *Framework Diamond* dapat membantu produktivitas dari Maulidan Games. Karena dengan adanya *pixel art converter* ini dapat memudahkan *designer* untuk membuat aset-aset

game berupa *pixel art* dengan efisien. Grafis *pixel art* juga diharapkan mampu menghemat penggunaan data karena penggunaan palet warna yang lebih sederhana.

II. METODOLOGI

Pada studi ini dibangun aplikasi *Pixel Art Converter* menggunakan algoritma klasterisasi *K-means*. Data masukan yang digunakan adalah citra aset karakter dari Maulidan Games. Data keluaran dari aplikasi ini adalah citra yang telah diproses menjadi *pixel art*.

Proses ini dimulai dengan memasukkan data citra aset karakter dari Maulidan Games. Kemudian dilakukan proses Segmentasi yang menggunakan algoritma dari *k-means clustering*. Citra aset karakter ini akan mengalami tahap *k-means clustering* untuk mendapatkan region-region warna pada citra yang ditentukan sesuai kluster *k-means*. Langkah terakhir adalah menjadikan tepian citra bergerigi atau *jaggy* yang merupakan ciri khas utama dari citra *pixel art*. **Gambar 1** berikut merupakan diagram alir keseluruhan proses *pixel art converter*.



Gambar 1. Diagram alir keseluruhan proses *Pixel Art Converter*

A. Segmentasi Citra Berbasis K-means Clustering

Clustering (klasterisasi) merupakan salah satu metode yang diterapkan secara luas dalam segmentasi citra dan statistik. Konsep utama *clustering* adalah menggunakan *centroid* untuk mewakili setiap *cluster* dan berdasarkan pada kesamaan dengan *centroid cluster* untuk dikelompokkan. Menurut karakteristik algoritma *clustering*, kita dapat secara kasar membagi ke dalam *clustering* "hirarkis" dan "partitional". [2] Secara umum algoritma dasar dari *K-means clustering* adalah sebagai berikut [3]:

1. Tentukan jumlah *cluster*
2. Alokasikan data ke dalam *cluster* secara random
3. Hitung *centroid*/rata-rata dari data yang ada di masing-masing *cluster*
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat. Perhitungan jarak terdekat menggunakan rumus jarak Euclidean di bawah ini:

$$r = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \quad (1)$$

5. Pembaruan nilai *centroid* dapat dilakukan dengan rumus sebagai berikut:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} q_k \quad (2)$$

dimana:

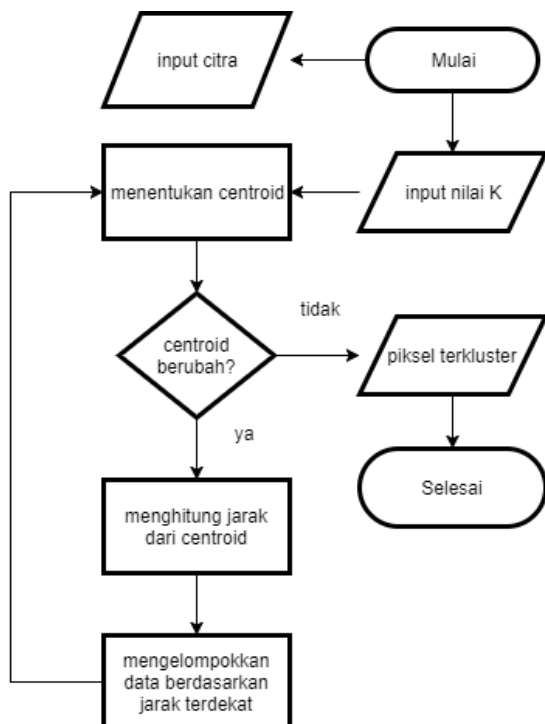
μ_k = titik centroid dari *cluster* ke-K

N_k = banyaknya data pada *cluster* ke-K

xq = data ke-q pada *cluster* ke-K

6. Kembali ke Step 3, apabila masih ada data yang berpindah *cluster* atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan.

Gambar 2 berikut merupakan diagram alir metode *k-means clustering*.



Gambar 2. Diagram alir *k-means clustering*

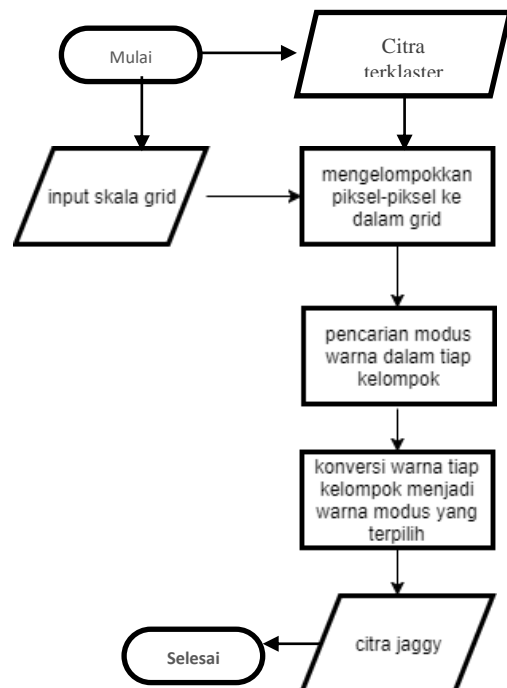
B. Pembuatan Tepian Jaggy

Tepian bergerigi merupakan ciri khas dari citra *pixel art*. Citra yang telah tersegmentasi akan diubah menjadi citra dengan tepian *jaggy*. Diperlukan variasi input skala grid yang digunakan untuk proses ini. Variasi yang diberikan mulai 10 hingga 100.

Tahap pertama dari pembuatan tepian *jaggy* ini dimulai dengan mengelompokkan piksel-piksel citra ke dalam skala yang telah diinputkan. Skala grid yang diinputkan menentukan jumlah piksel tiap kelompok grid. Cara mendapatkan jumlah piksel dalam kelompok grid adalah ukuran citra dibagi dengan input skala grid tepian. Jika input yang dipilih adalah 80 dan ukuran citra adalah 320 x 320 piksel, berarti dalam satu kelompok piksel terdiri dari 4 x 4 piksel.

Tahap kedua yaitu tiap kelompok piksel citra ditentukan warna yang dominan. Yaitu dengan cara menentukan frekuensi setiap warna yang ada pada setiap grid. Kemudian frekuensi tersebut diurutkan secara *descending*, urutan tersebut menyatakan warna dominan yang terpilih.

Warna dominan yang terpilih akan digunakan untuk mengubah seluruh warna dalam kelompok piksel. Hal ini dilakukan untuk semua kelompok piksel citra pada input yang telah terkluster. Cara ini dapat mengubah bentuk citra memiliki tepian yang terkotak-kotak atau *jaggy*. **Gambar 3** berikut merupakan diagram alir pembuatan tepian *jaggy*.



Gambar 3. Diagram alir pembuatan tepian *jaggy*

III. UJI COBA DAN EVALUASI

Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan. Citra yang digunakan dalam aplikasi ini berformat .jpg atau .png. Ukuran citra memiliki lebar dan panjang yang sama). Untuk menguji aplikasi *pixel art converter* diperlukan 10 citra asset karakter *game* pada Maulidan Games dengan variasi ukuran dan jenis yang berbeda.



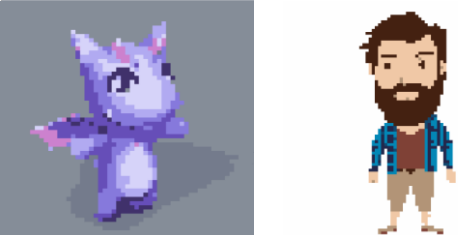
A. Uji Coba dengan Variasi Ukuran Citra Input

Uji coba dengan variasi ukuran digunakan untuk mengetahui pengaruh ukuran terhadap hasil klastering *k-means* dan citra hasil akhir *pixel art*. Pengujian terhadap ukuran citra input 90 x 90 piksel, 270 x 270 piksel, dan 512 x 512 piksel.

Parameter lain yang digunakan untuk uji coba variasi input ini seperti skala grid adalah sebesar 80 skala dan nilai K sebesar 8.

Contoh hasil dari uji coba variasi ukuran citra input dapat dilihat pada **Tabel 1**. Dari hasil uji coba tersebut dapat dilihat bahwa untuk citra ukuran 90 x 90 piksel menghasilkan citra dengan tepian *jaggy* yang kurang terlihat. Citra untuk ukuran 270 x 270 piksel dan 512 x 512 piksel menghasilkan citra *jaggy* yang baik.

Tabel 1.
Tabel hasil uji coba variasi ukuran input

Ukuran Citra	Output Citra
90 x 90	
270 x 270	
512 x 512	

B. Uji Coba dengan Variasi Nilai K untuk K-means Clustering

Variasi nilai K dilakukan untuk mengetahui pengaruh nilai K tersebut terhadap hasil citra yang dilakukan *clustering* dan output citra *pixel art*. Variasi input nilai K yang digunakan adalah 4, 8, 16, dan 25.




Parameter lain yang digunakan untuk uji coba variasi skala grid ini seperti ukuran citra input adalah ukuran masing-masing citra pada kelompok 3, yaitu 512 x 512 dan skala grid yang digunakan adalah 80.

Hasil uji coba dengan variasi nilai K ditunjukkan pada **Tabel 2**. Pada hasil uji coba tersebut dapat disimpulkan bahwa nilai K berpengaruh pada jumlah variasi warna citra hasil. Semakin tinggi nilai K, semakin banyak pula warna citra hasil yang ditampilkan.

Nilai K kecil berguna untuk membuat warna *gradient* yang terlalu rumit menjadi sederhana, sehingga dapat membuat citra *pixel art* yang baik. Namun, nilai K kecil memiliki kelemahan yaitu, adanya bagian citra yang kurang detail. Pada citra pertama ditunjukkan bahwa dengan K

bernilai 4, 8, dan 16 tidak muncul hidung dari citra karakter, sedangkan dengan K bernilai 25 dapat terlihat hidung dari citra karakter tersebut. Selain itu, pada citra kedua juga menunjukkan hasil K = 4 detail makanan yang dipegang oleh citra karakter koki kurang jelas, namun pada nilai K 8, 16, dan 25 terlihat jelas.

Tabel 2.
Tabel hasil uji coba variasi nilai K

Skala Warna	Output
Hasil K = 4	
Hasil K = 8	
Hasil K = 16	
Hasil K = 25	

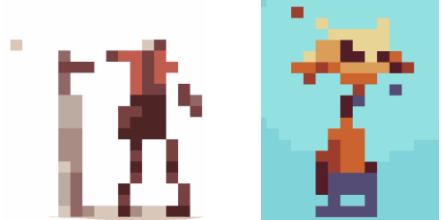


C. Uji Coba dengan Variasi Skala Grid untuk Membuat Tepian Jaggy

Variasi skala grid dilakukan untuk mengetahui pengaruh skala grid terhadap output citra *pixel art*. Variasi input skala yang digunakan adalah 20 (menyatakan skala yang sangat besar), 55 (menyatakan skala yang normal), dan 100 (menyatakan skala yang sangat kecil).

Parameter lain yang digunakan untuk uji coba variasi skala grid ini seperti ukuran citra input adalah ukuran masing-masing citra pada kelompok 3, yaitu 512 x 512 dan nilai K sebesar 8.

Hasil uji coba dapat dilihat pada **Tabel 3**. Hasil uji coba dengan variasi nilai skala grid menunjukkan bahwa semakin tinggi nilai skala, bentuk kotak-kotak citra hasil akan semakin kecil, dan tepian akan semakin halus atau mendekati bentuk asli.

Tabel 3.
Tabel hasil uji coba variasi nilai skala grid

Ukuran Skala Grid	Output Citra
Hasil skala 20	
Hasil skala 55	
Hasil skala 100	

D. Uji Coba Kebergunaan

Pengujian kebergunaan ini dilakukan untuk mengetahui tingkat kebergunaan aplikasi terhadap pengguna. Pengujian dilakukan memberikan kesempatan kepada CMO (*Credit Marketing Officer*) dari Maulidan Games, yang bernama Muhammad Hildi Radya Nararya, untuk mencoba aplikasi *Pixel Art Converter*. Setelah pengujian aplikasi *Pixel Art Converter*, pengguna akan diminta untuk mengisi kuesioner untuk mengetahui tanggapan dari pengguna.

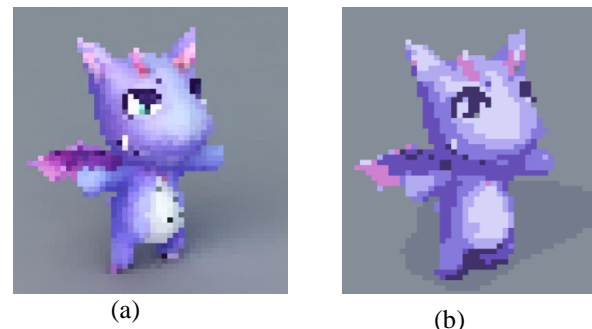
Berdasarkan pengujian yang dilakukan, aplikasi ini dinyatakan telah dapat digunakan oleh tim desain Maulidan Games. Input dan proses, serta hasil output yang didapat telah sesuai dengan kebutuhan yang dibutuhkan. Output citra *pixel art* sudah sesuai dengan gaya *pixel art* yang digunakan oleh para *designer*.

E. Evaluasi

Hasil uji coba *pixel art converter* dengan variasi ukuran input dapat menunjukkan bahwa citra hasil memiliki hasil *pixel art* yang optimal dengan ukuran yang relatif besar. Ukuran yang dapat digunakan yaitu antara 270 x 270 piksel dan 512 x 512 seperti yang ditunjukkan pada uji coba. Citra yang memiliki ukuran kecil, seperti yang ditunjukkan pada uji coba yaitu 90, menghasilkan citra *pixel art* yang kurang baik. Citra ukuran kecil tersebut, terlihat kabur, kotak-kotak atau *jaggy* yang kurang terlihat, sehingga tidak dapat dimanfaatkan oleh tim desainer Maulidan Games.

Citra dengan jenis desain *gradient color* tidak dapat langsung diubah menjadi bentuk *pixel art* karena jenis warna yang rumit. Warna tersebut harus disederhanakan dengan segmentasi berbasis *k-means clustering*. Hasil segmentasi tersebut dapat diubah menyerupai citra dengan jenis *flat color*. Sehingga, citra *gradient color* yang telah tersegmentasi tersebut dapat mendapat hasil *pixel art* yang baik. Berikut contoh perbandingan citra *gradient color* yang

di-segmentasi terlebih dahulu sebelum menjadi *pixel art* dan citra tanpa segmentasi.



Gambar 4. Perbandingan citra output (a) tanpa segmentasi (b) dengan segmentasi

Citra pada **Gambar 4 (a)** menunjukkan bahwa tanpa dilakukan segmentasi berbasis *k-means clustering*, citra hasil *pixel art* memuat warna *gradient* yang rumit dan bagian mulut citra karakter terlihat rusak. Berbeda dengan hasil segmentasi berbasis *k-means clustering* pada **Gambar 4 (b)** memiliki hasil warna lebih sederhana seperti *flat color* dan bagian mulut citra karakter terlihat lebih rapi.

Pixel art memiliki ciri khas bentuk tepian bergerigi/*jaggy*. Pada *pixel art converter* yang dibangun dipengaruhi oleh input skala grid oleh pengguna. Pada uji coba yang dilakukan dengan skenario variasi input skala grid, dapat ditunjukkan bahwa semakin tinggi nilai skala, hasil *pixel art* semakin halus atau menyerupai bentuk asli.

Terdapat beberapa nilai optimal yang dapat digunakan oleh tim desain Maulidan Games supaya mendapatkan hasil citra *pixel art* yang baik. Berikut merupakan nilai-nilai optimal yang diberikan oleh CMO Maulidan Games,

1. Untuk citra *gradient color* nilai optimal untuk skala warna atau nilai K untuk *K-means* adalah 25, sedangkan untuk citra *flat color* menggunakan nilai K 16.
2. Nilai skala grid untuk membuat tepian adalah 80.
3. Ukuran citra yang baik untuk dilakukan konversi menjadi piksel art adalah 100 x 100 piksel.

Namun, perlu adanya pengembangan, yaitu:

1. Memberikan *range* skala grid dengan pilihan: 8, 16, 32, 64, 128, 256.
2. Diberikan indikator (seperti *loading screen*) yang menyatakan sedang dilakukan proses konversi.
3. Memberikan pilihan modifikasi warna RYB/RGB dari citra hasil konversi, sehingga apabila pengguna dapat mengubah menjadi *style* warna 8 bit.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan studi ini dapat ditarik kesimpulan bahwa:

1. Rancang bangun *pixel art converter* dapat dibangun dengan segmentasi berdasarkan *k-means clustering* dengan cara memberikan input citra dan nilai K yang akan mempengaruhi warna citra *pixel art* yang dihasilkan.
2. Nilai skala grid untuk membuat tepian *jaggy* mempengaruhi hasil citra *pixel art*. Semakin besar nilai skala grid, citra hasil *pixel art* akan semakin halus atau mendekati bentuk asli citra inputnya dan semakin kecil nilai skala grid, bentuk kotak-kotak (tepian *jaggy*) akan semakin besar dan menjauhi bentuk asli dari citra input.

3. Nilai K untuk segmentasi berbasis *k-means clustering* mempengaruhi warna citra hasil *pixel art*. Semakin tinggi nilai K, maka jumlah variasi warna pada citra hasil *pixel art* semakin banyak.
4. Segmentasi berbasis *k-means clustering* dapat mengubah citra dengan jenis *gradient color* menyerupai *flat color* sehingga dapat memberikan citra hasil *pixel art* yang lebih baik.
5. Ukuran citra mempengaruhi kualitas citra hasil *pixel art*. Semakin tinggi ukuran citra input, kualitas citra hasil *pixel art* juga akan semakin baik.
6. Citra hasil *pixel art* yang optimal didapatkan dari input:
 - a. Untuk citra *gradient color* nilai optimal untuk skala warna atau nilai K untuk *K-means* adalah 25, sedangkan untuk citra *flat color* menggunakan nilai K 16.
 - b. Nilai skala grid untuk membuat tepian adalah 80.

- c. Ukuran citra yang baik untuk dilakukan konversi menjadi *pixel art* adalah 100 x 100 piksel.

B. Saran

Berikut saran yang diberikan untuk studi ini:

1. Dapat dikembangkan fitur-fitur lain, seperti menyimpan gambar dan menghilangkan background citra karakter
2. Perlu adanya analisa parameter lain yang dapat digunakan untuk memperbaiki *pixel art converter*

DAFTAR PUSTAKA

- [1] R. Prayuga, "Apakah itu Pixel Art?," 2015. [Online]. Available: <http://inovasipintar.com/apakah-itu-pixel-art/>.
- [2] G. Shir, "Muzk," 2017. [Online]. Available: <https://medium.muz.li/why-gradients-are-the-new-colors-3d8d42a7a6fc>.
- [3] A. I. and A. L. I. Burmistrov, T. Zlokazova, "Flat Design vs Traditional Design: Comparative Experimental Study," *Flat Des. vs Tradit. Des. Comp. Exp. Study*, pp. 107–108.